



# scRNA-seq Snap pipeline: Hands-on

---

Asha Jacob Jannu

Bioinformatics Core  
Department of Developmental Neurobiology



**SC RNA**  
**SEQ SNAP**



# Snap Pipeline: Quick Start Guide

---



## DNB Training

---

### scRNA-seq Snap Pipeline – Quick Start Guide

---

This is a step-by-step quick reference for setting up, configuring, and running the `sc-rna-seq-snap` pipeline on an HPC environment. This guide covers essential commands, file organization, and job submission, tailored for use with `RStudio/R v4.4.0` and `Seurat v4.4.0`.

Please execute the steps **in the specified order** to ensure a successful Snap pipeline run.

For best practices and detailed guidelines on effectively using the Snap pipeline, please review the [Tutorial and documentation for the snap pipeline](#).

---

### Workflow Overview

---

- 1. HPC Login & Basic Setup – connect to the cluster and create your project folder
- 2. Fork & Clone Repository – copy the training repo to your GitHub and HPC
- 3. Setup Singularity Container – prepare your analysis environment
- 4. Prepare Metadata File – organize input sample information
- 5. Configure YAML File – edit file paths and project details
- 6. Run Modules – execute the training analysis step by step

[Link to Quick Start Guide](#)



# Bioinformatics Core Team at DNB

---



**Cody Ramirez, PhD**

Director of Bioinformatics Core  
Sr. Bioinformatics Research Scientist  
Boston, Massachusetts



**Antonia Chroni, PhD**

Sr. Bioinformatics Research Scientist  
New York, New York



**Sharon Freshour, PhD**

Bioinformatics Research Scientist  
Tacoma, Washington



**Asha Jacob Jannu, MS**

Bioinformatics Research Scientist  
Indianapolis, Indiana

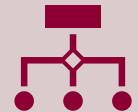


# What we'll do today

---



Run Snap on a small 10x scRNA-seq dataset



Quick overview of results (QC → alignment → clustering → integration → annotation)



Explore results with R Shiny



You'll leave with: a working folder containing results, reports, an R Shiny app, and a quick start guide



# Single-cell RNA-seq answers..

- **What cell types are present?**  
UMAP colored by clusters; marker table → assign labels
- **Are there rare populations?**  
Cluster sizes; marker enrichment for small clusters
- **What states are cells in (cycling, activated, stressed)?**  
Feature plots for state markers; QC/state scores
- **How similar/different are samples?**  
Integrated UMAP; batch overlay; sample composition bars
- **Which genes define each type?**  
Top markers per cluster; heatmaps; volcano tables



# To answer those questions, you need a chain of steps...

---

Raw sequencing data QC → Alignment & Quantification → Cell QC/ filtering → Contamination removal →  
Normalization → Integration → Clustering → Annotation → Differential Expression →  
Pathways/Trajectories → Reporting & Sharing

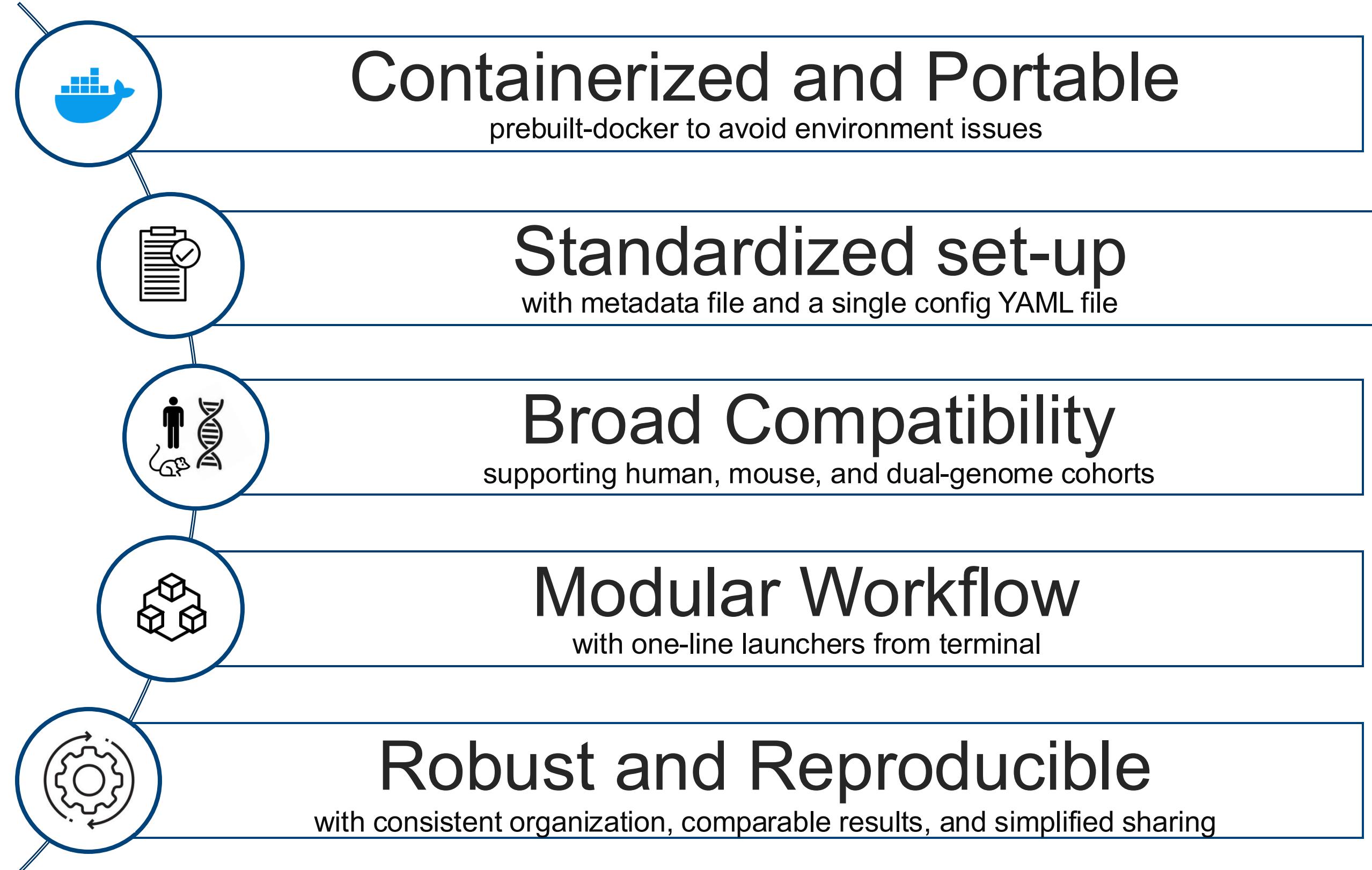


# **... and each step has multiple tools**

---

FASTQC / Cell Ranger / Seurat / scDblFinder / SoupX / Harmony / LIGER/ R Shiny ...





# Snap-seq pipeline on GitHub

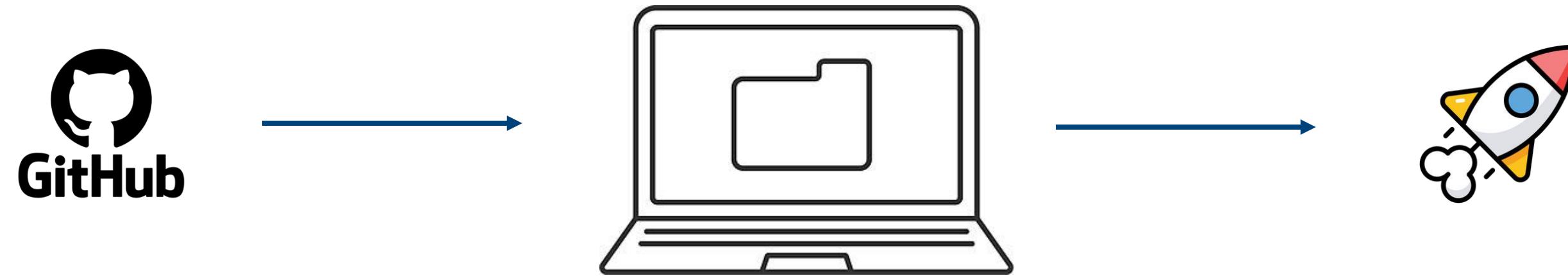
---

<https://github.com/stjude-dnb-binfcore/sc-rna-seq-snap>



# Fork & Clone, Prep and Run

---



# Quick Start Guide to Snap Pipeline

---

## What to look for:

- 👉 **Pipeline flow:** FastQC → Cellranger → Upstream analysis → Integrative analysis → Cluster cell analysis → Cell type Annotation → R Shiny
- 🚫 **No skipping** or jumping ahead. Later steps depend on earlier outputs.
- 📁 **Outputs:** analysis/, plots/, reports/ : analysis/<module>/, results/, plots/
- ⌚ **Timing:** Runtimes apply only to the training dataset

## Success checks after each step:

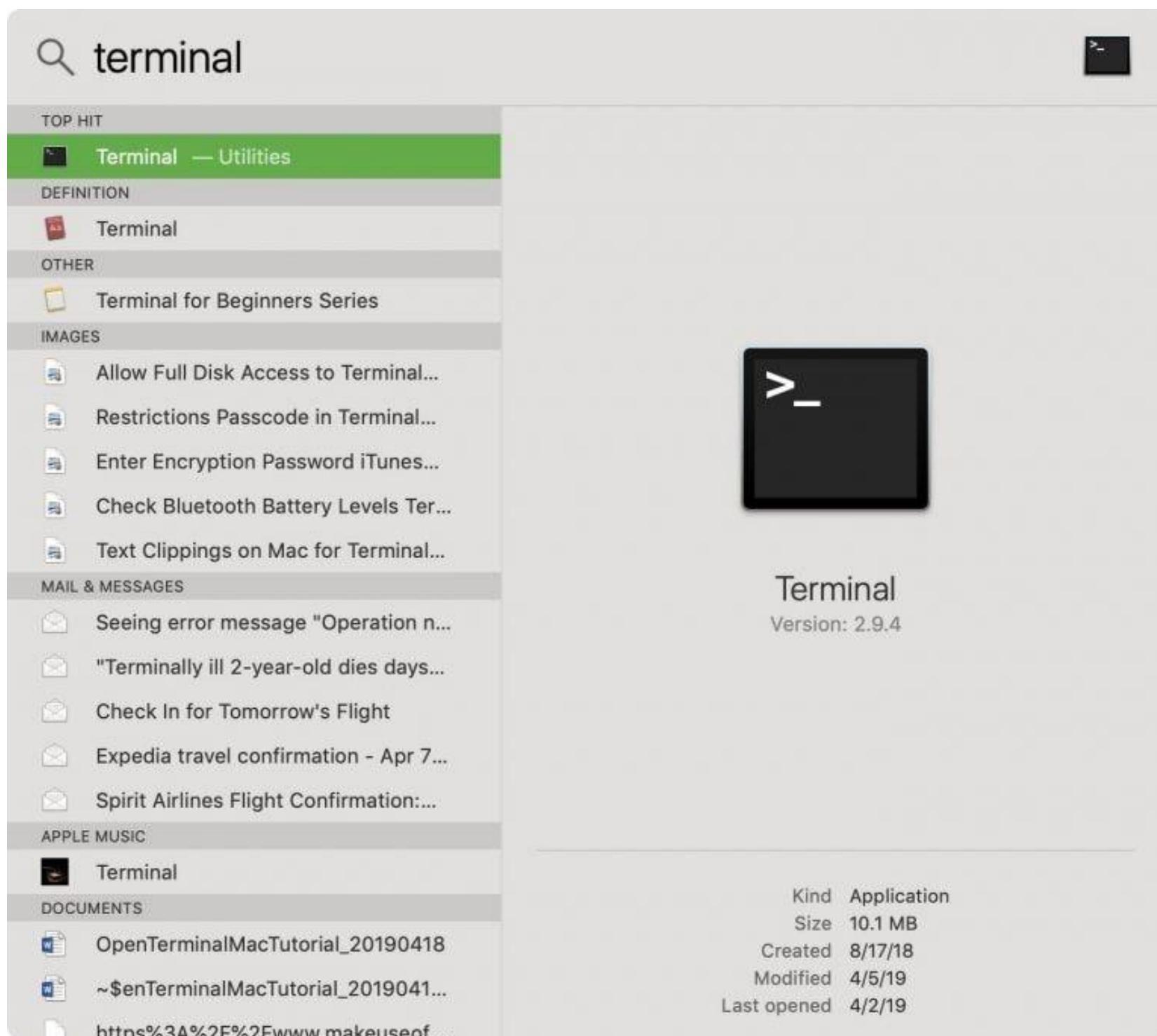
- 🧠 **Check job logs** - Open job.out and job.err to confirm success / inspect errors
- 📁 **Verify key outputs** - Look for expected artifacts (e.g., web\_summary.html, Reports, UMAPs, marker tables, R shiny)
- 🔑 **3. Double-check file paths** - If something seems off, run `pwd`, `ls -lh` to confirm directories



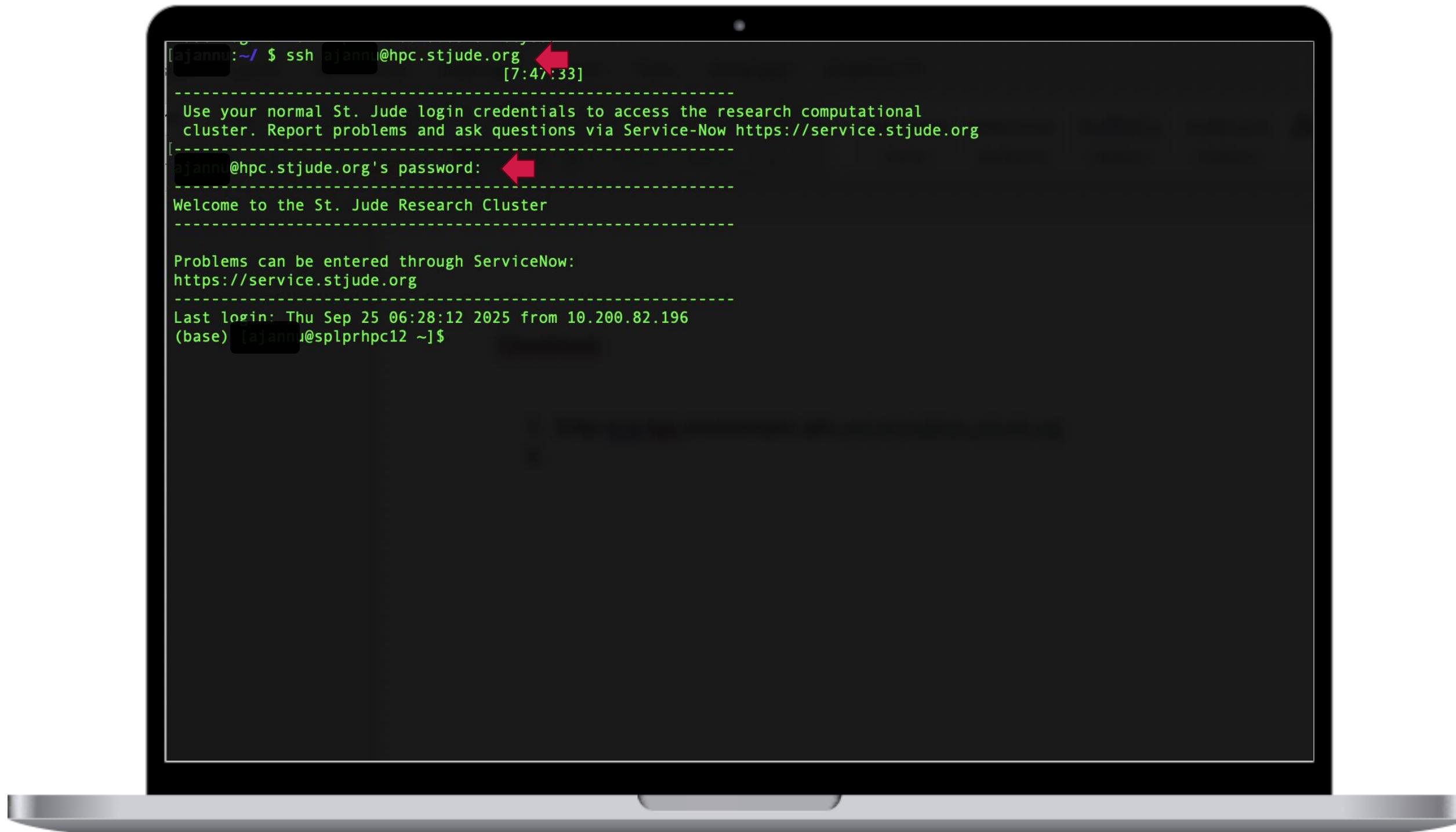
# Let's Begin



# Search: Terminal (Mac) or Windows Terminal (Windows)

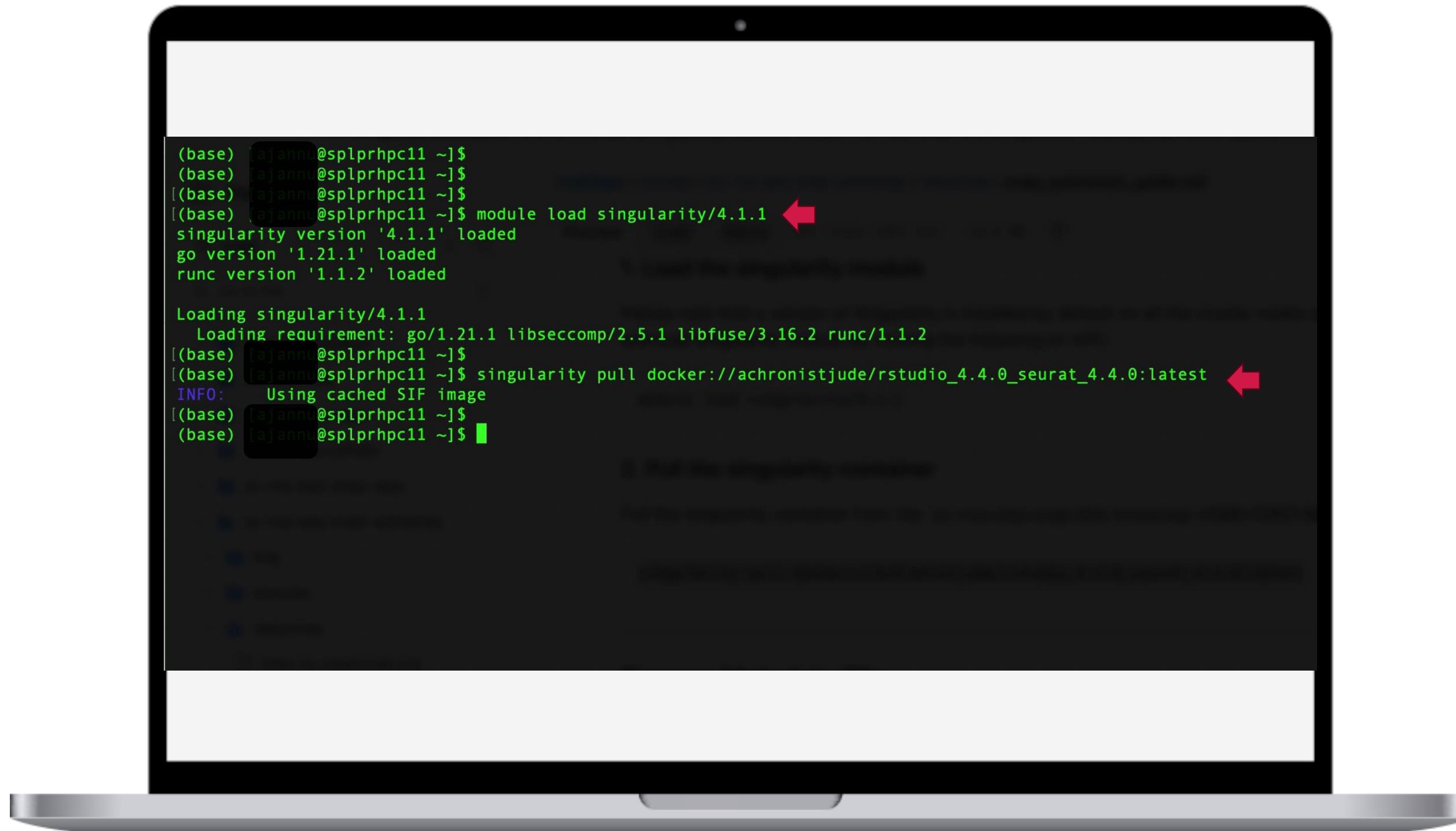


# Login to HPC



# Setup Singularity Container

Runtime: 20mins (First-time setup) – 5mins (typical time)

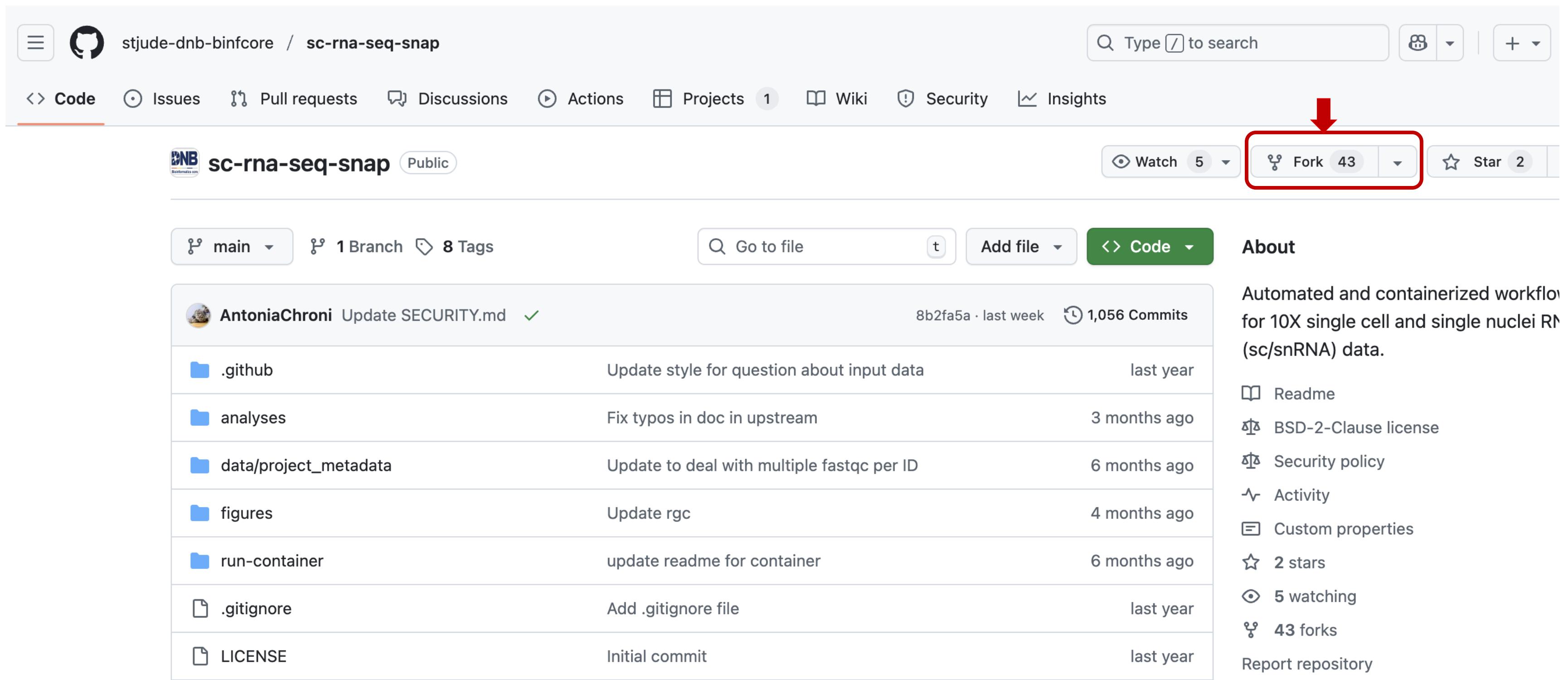


# Fork & Clone, Prep and Run

---



# Setup Snap Pipeline Locally



The screenshot shows a GitHub repository page for 'sc-rna-seq-snap'. The top navigation bar includes links for Code, Issues, Pull requests, Discussions, Actions, Projects (1), Wiki, Security, and Insights. A search bar and user profile icons are also present. Below the header, the repository name 'sc-rna-seq-snap' is shown as Public. The main content area displays the repository's code structure, showing branches ('main'), tags ('8 Tags'), and a list of commits. A red box and arrow highlight the 'Fork' button, which has 43 forks. To the right, there is an 'About' section with a detailed description of the repository, including its purpose as an automated and containerized workflow for 10X single cell and single nuclei RNA (sc/snRNA) data. Other repository details listed include Readme, BSD-2-Clause license, Security policy, Activity, Custom properties, 2 stars, 5 watching, 43 forks, and a 'Report repository' link.

stjude-dnb-binfcore / sc-rna-seq-snap

Type / to search

Code Issues Pull requests Discussions Actions Projects 1 Wiki Security Insights

sc-rna-seq-snap Public

Watch 5 Fork 43 Star 2

main 1 Branch 8 Tags

Go to file Add file Code

**About**

Automated and containerized workflow for 10X single cell and single nuclei RNA (sc/snRNA) data.

Readme

BSD-2-Clause license

Security policy

Activity

Custom properties

2 stars

5 watching

43 forks

Report repository

Commit	Message	Date
AntoniaChroni	Update SECURITY.md	8b2fa5a · last week
.github	Update style for question about input data	last year
analyses	Fix typos in doc in upstream	3 months ago
data/project_metadata	Update to deal with multiple fastqc per ID	6 months ago
figures	Update rgc	4 months ago
run-container	update readme for container	6 months ago
.gitignore	Add .gitignore file	last year
LICENSE	Initial commit	last year



# Setup Snap Pipeline Locally

## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*



/ sc-rna-seq-snap-dnb-tr

sc-rna-seq-snap-dnb-training is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

### Description

dnb-training-2025-10-29

23 / 350 characters

Copy the main branch only

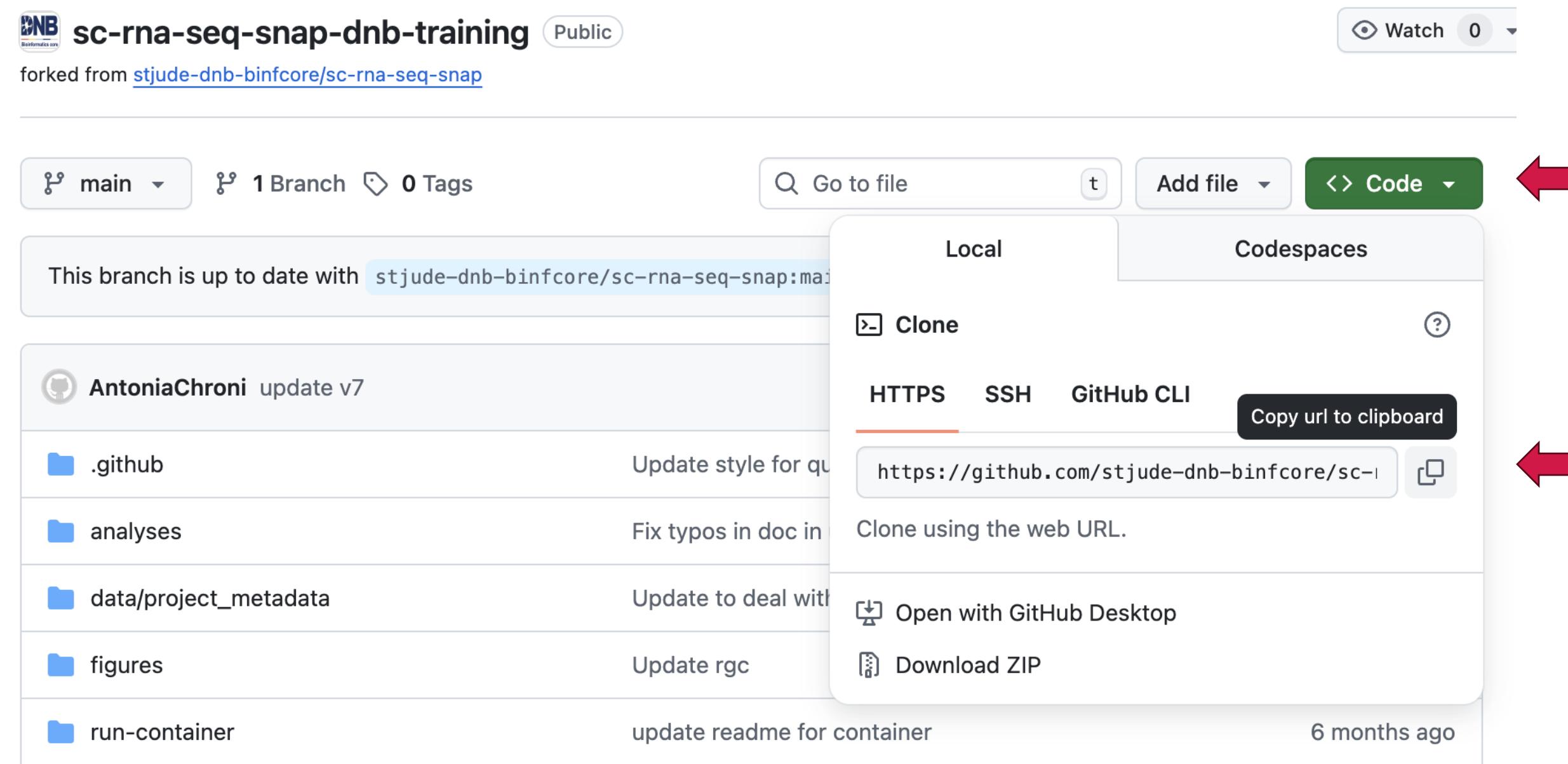
Contribute back to stjude-dnb-binfcore/sc-rna-seq-snap by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

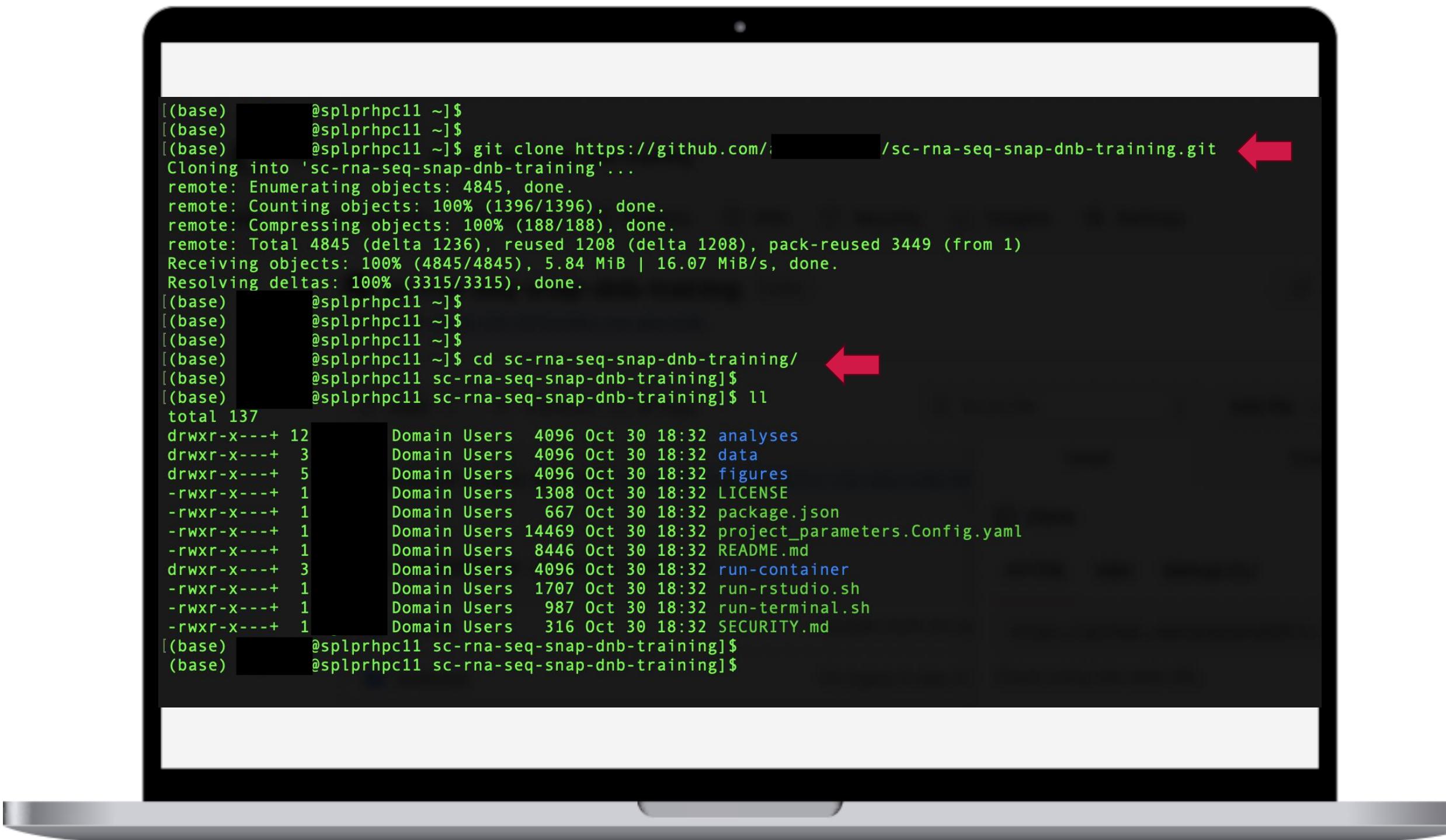
Create fork



# Clone Repository



# Clone the Repository

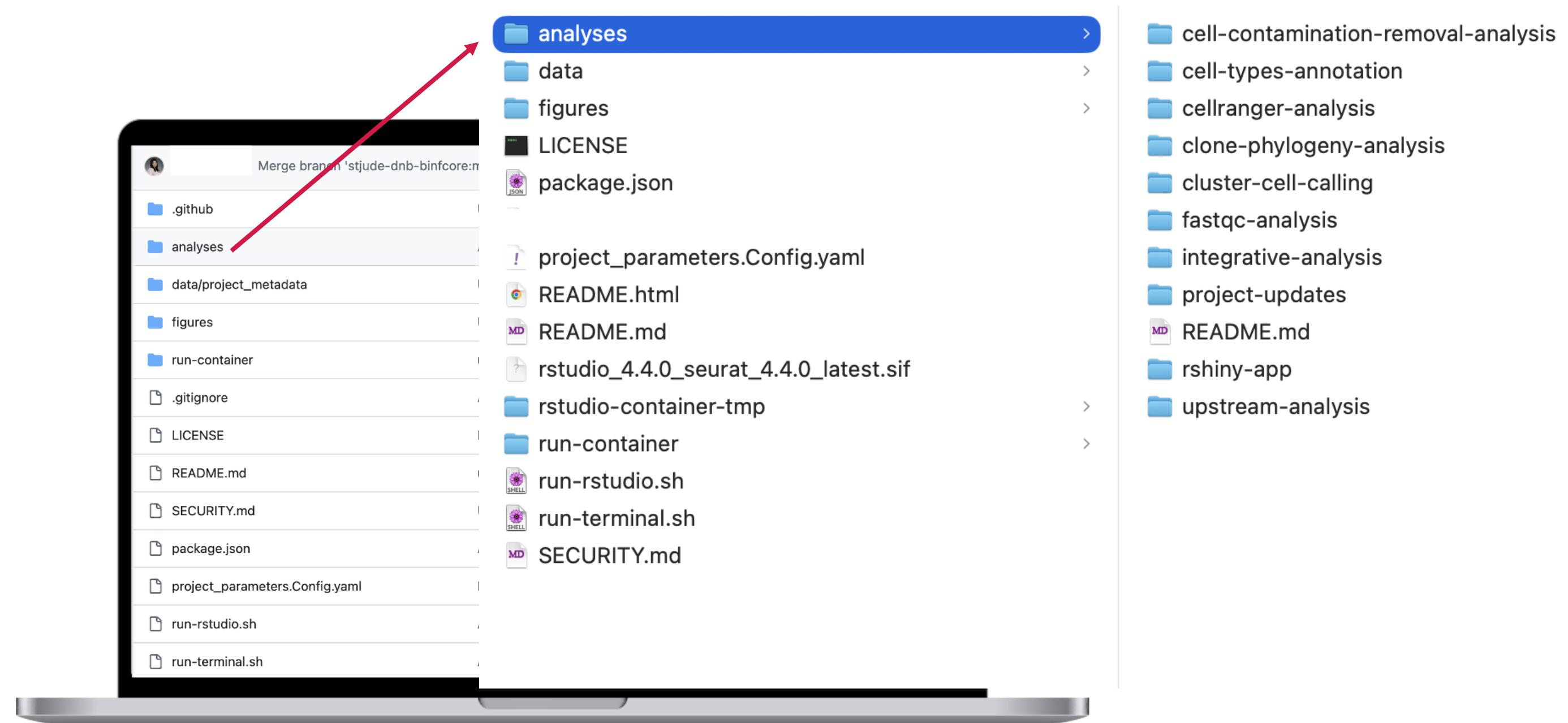


```
[base] @splprhpc11 ~]$  
[base] @splprhpc11 ~]$  
[base] @splprhpc11 ~]$ git clone https://github.com/ [REDACTED] /sc-rna-seq-snap-dnb-training.git  
Cloning into 'sc-rna-seq-snap-dnb-training'... ←  
remote: Enumerating objects: 4845, done.  
remote: Counting objects: 100% (1396/1396), done.  
remote: Compressing objects: 100% (188/188), done.  
remote: Total 4845 (delta 1236), reused 1208 (delta 1208), pack-reused 3449 (from 1)  
Receiving objects: 100% (4845/4845), 5.84 MiB | 16.07 MiB/s, done.  
Resolving deltas: 100% (3315/3315), done.  
[base] @splprhpc11 ~]$  
[base] @splprhpc11 ~]$  
[base] @splprhpc11 ~]$  
[base] @splprhpc11 ~]$ cd sc-rna-seq-snap-dnb-training/ ←  
[base] @splprhpc11 sc-rna-seq-snap-dnb-training]$  
[base] @splprhpc11 sc-rna-seq-snap-dnb-training]$ ll  
total 137  
drwxr-x---+ 12 Domain Users 4096 Oct 30 18:32 analyses  
drwxr-x---+ 3 Domain Users 4096 Oct 30 18:32 data  
drwxr-x---+ 5 Domain Users 4096 Oct 30 18:32 figures  
-rwxr-x---+ 1 Domain Users 1308 Oct 30 18:32 LICENSE  
-rwxr-x---+ 1 Domain Users 667 Oct 30 18:32 package.json  
-rwxr-x---+ 1 Domain Users 14469 Oct 30 18:32 project_parameters.Config.yaml  
-rwxr-x---+ 1 Domain Users 8446 Oct 30 18:32 README.md  
drwxr-x---+ 3 Domain Users 4096 Oct 30 18:32 run-container  
-rwxr-x---+ 1 Domain Users 1707 Oct 30 18:32 run-rstudio.sh  
-rwxr-x---+ 1 Domain Users 987 Oct 30 18:32 run-terminal.sh  
-rwxr-x---+ 1 Domain Users 316 Oct 30 18:32 SECURITY.md  
[base] @splprhpc11 sc-rna-seq-snap-dnb-training]$  
[base] @splprhpc11 sc-rna-seq-snap-dnb-training]$
```

[Documentation-snap-repo-tutorial](#)

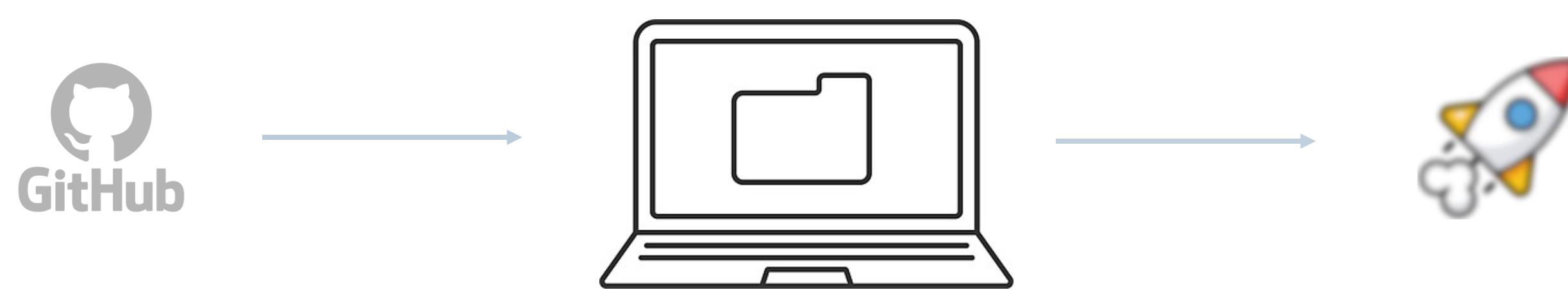


# Folder Structure



# Fork & Clone, Prep and Run

---



# Hands-On Checklist: • Metadata • Data • Config • Terminal



DATA



METADATA FILE

	A	B	C	D	E
1	ID	SAMPLE	FASTQ	condition	tissue_type
2	sample1	seq_submiss	/absolute_pa	WT	Retina
3	sample2	seq_submiss	/absolute_pa	Knock-out	Retina

```
! project_parameters.Config.yaml ×
Users > ajannu > Downloads > ! project_parameters.Config.yaml
1 # the following parameters are the same across the project and might be needed in more than one module #
2 root_dir: "/sc-rna-seq-snap" # Absolute path to the main dir of the project where Github repo lives
3 data_dir: "/sc-rna-seq-snap/analyses/cellranger-analysis/results/02_cellranger_count/DefaultParameters" # Absolute path to data dir
4 metadata_dir: "/sc-rna-seq-snap/data/project_metadata" # Absolute path to metadata dir of the project. File name always named as: `project_metadata.tsv` or `project_metadata.csv`
5 metadata_file: "project_metadata.tsv" # Options: "project_metadata.tsv" (default) or name as user wants. It needs to be in `tsv` format
6 genome_name: "GRCm39" # define genome reference and versioning. Options: (1) human: "GRCh38", "hg19", and "GRCh38_GFP_tdTomato"; (2) mouse: "GRCm39"
7 PROJECT_NAME: "PROJECT_NAME"
8 PI_NAME: "PI_NAME"
9 TASK_ID: "NA"
10 PROJECT_LEAD_NAME: "NA"
11 DEPARTMENT: "Developmental Neurobiology"
12 LEAD_ANALYSTS: "Antonia Chroni, PhD"
13 GROUP_LEAD: "Cody A. Ramirez, PhD"
14 CONTACT_EMAIL: "antonia.chroni@stjude.org"
15 PIPELINE: "Standard sc-/sn-RNA-Seq Analysis in 10X Genomics data"
16 START_DATE: "10/15/2024"
17 COMPLETION_DATE: "ONGOING"
18
19 # the following parameters are set up as default values and/or are specific for the following modules:
20 # `./analyses/fastqc-analysis`
21 # FASTQ paths to the fastqc files with format: `path1/*R2*.fastq.gz` are extracted from the `metadata_dir`.
22 # No need to manually define variables
23
24 # `./analyses/cellranger-analysis`
25 genome_reference_path: "./" # Absolute path to genome reference to be used for the `cellranger-analysis` module
26 cellranger_parameters: "DefaultParameters" # Options: "DefaultParameters", "ForcedCells8000Parameters", or else
27 genome_name_cellranger: "GRCm39" # define the genome of preference for dual genomes. In case for single genomes, please use the same
28 create_bam_value: "true" # Options: "true" (default) or "false". For non-human experiments, we recommend setting this to 'false' to
29 # Define the sample ID prefix(es) used in this project.
30 # Sample IDs should follow a format like: PREFIX001 (e.g., DYE001, ABC002).
31 # You can specify multiple prefixes if your project uses more than one.
32 sample_prefix:
33   - DYE
34   - ABC-
35   - XYZ_
36
37 # `./analyses/upstream-analysis`
38 print_pdf_seurat_multiple_samples: "YES" # Options: "YES" (default ALWAYS), for `02B_run_seurat_qc_multiple_samples.R`
39 use_condition_split_seurat_multiple_samples: "NO" # Options: "NO" (default ALWAYS), for `02B_run_seurat_qc_multiple_samples.R`
40 grouping: "orig.ident" # define grouping to use
41 Regress_Cell_Cycle_value: "NO" # Options: "YES", "NO", "DIFF", OR "mtDNA". Indicates whether or not to regress for cell cycle and, i
```

CONFIG FILE

TERMINAL



# Quick Quiz

---

**WHICH COMMAND LISTS ALL FILES IN DIRECTORY?**

- A. cd
- B. ls

**WHAT DOES “cd” – DO?**

- A. Moves you from one directory to another
- B. Copy and paste



# Fork & Clone, Prep and Run

---



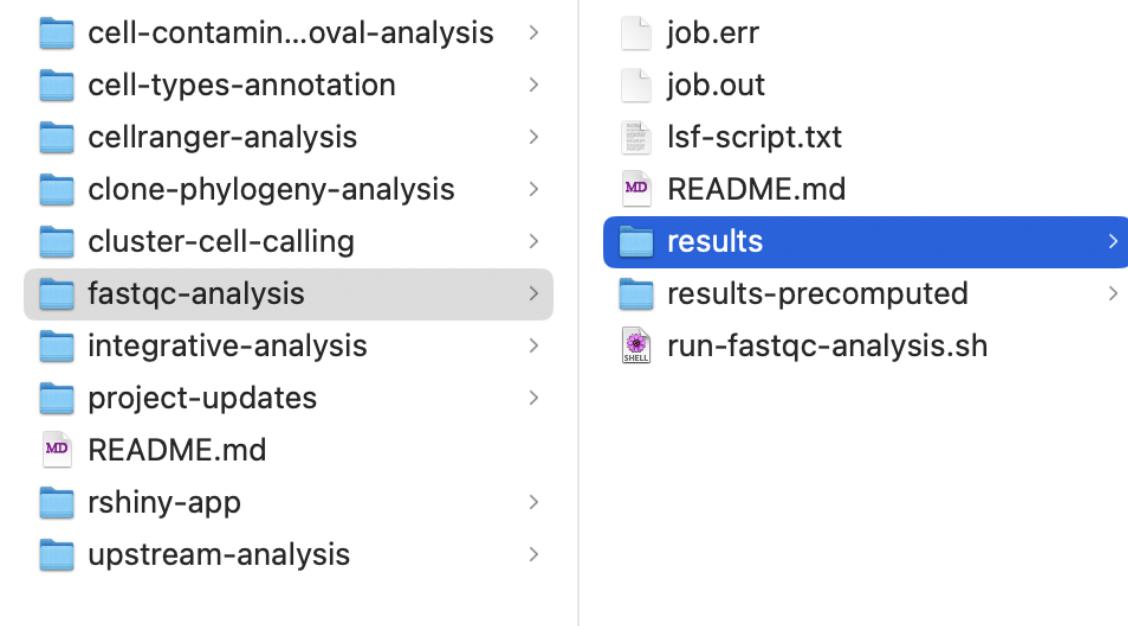
# 1. Navigate to the analyses folder

```
total 6297096
drwx-----+ 12          Domain Users      4096 Sep 16 09:35 analyses
drwx-----+ 3          Domain Users      4096 Sep 16 09:12 data
drwx-----+ 5          Domain Users      4096 Sep 16 08:57 figures
-rwx-----+ 1          Domain Users     1308 Sep 16 08:57 LICENSE
-rwx-----+ 1          Domain Users      667 Sep 16 08:57 package.json
-rwx-----+ 1          Domain Users    15152 Oct 27 20:19 project_parameters.Config.yaml
-rwx-----+ 1          Domain Users   2007616 Oct  7 11:27 README.html
-rwx-----+ 1          Domain Users      8446 Oct  7 11:27 README.md
-rwx-----+ 1          Domain Users    2833 Feb 24 2022 retinal_cell_type_top50_gene_markers.tsv
-rwxr-xr-x+ 1          Domain Users  6445817856 Sep 16 09:26 rstudio_4.4.0_seurat_4.4.0_latest.sif
drwx-----+ 3          Domain Users      4096 Sep 16 09:25 rstudio-container-tmp
drwx-----+ 3          Domain Users      4096 Oct  6 07:08 run-container
-rwx-----+ 1          Domain Users     1707 Sep 16 08:57 run-rstudio.sh
-rwx-----+ 1          Domain Users      987 Sep 16 08:57 run-terminal.sh
-rwx-----+ 1          Domain Users      619 Sep 16 08:57 SECURITY.md
[(base)  ]@splprhpc08 sc-rna-seq-snap-dnb-training]$
[(base)  ]@splprhpc08 sc-rna-seq-snap-dnb-training]$ cd analyses/
[(base)  ]@splprhpc08 analyses]$ ll
total 41
drwx-----+ 2          Domain Users      4096 Sep 16 08:57 cell-contamination-removal-analysis
drwx-----+ 5          Domain Users     8192 Oct 27 14:59 cellranger-analysis
drwx-----+ 7          Domain Users     8192 Oct 27 15:01 cell-types-annotation
drwx-----+ 2          Domain Users      4096 Sep 16 08:57 clone-phylogeny-analysis
drwx-----+ 8          Domain Users      4096 Oct 27 15:00 cluster-cell-calling
drwx-----+ 4          Domain Users      4096 Oct 27 14:59 fastqc-analysis
drwx-----+ 7          Domain Users      4096 Oct 27 15:00 integrative-analysis
drwx-----+ 3          Domain Users      4096 Oct 27 21:11 project-updates
-rwx-----+ 1          Domain Users     2427 Sep 16 08:57 README.md
drwx-----+ 5          Domain Users      4096 Sep 27 22:18 rshiny-app
drwx-----+ 9          Domain Users      4096 Oct 27 14:59 upstream-analysis
(base)  ]@splprhpc08 analyses]$
```



# 2. `fastqc-analysis` module

Runtime: 10mins (training dataset) – 3hrs (8 samples; 50,000 cells)



```
(base) [splprhpc11 analyses]$ cd fastqc-analysis/
(base) [splprhpc11 fastqc-analysis]$ ls
total 3
-rwxr-x---+ 1 Domain Users 230 Oct 28 17:55 lsf-script.txt
-rwxr-x---+ 1 Domain Users 3418 Oct 28 17:55 README.md
-rwxr-x---+ 1 Domain Users 3580 Oct 28 17:55 run-fastqc-analysis.sh
(base) [aianni@splprhpc11 fastqc-analysis]$ bsub < lsf-script.txt
Job <272013689> is submitted to default queue <standard>.
(base) [splprhpc11 fastqc-analysis]$ bjobs
JOBID      USER      STAT  QUEUE   FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
271835463  ;        RUN    interactive svlp pondeman 2*noderome1 *OMEM_48hr Oct 27 10:56
272013689  ;        PEND   standard   splprhpc11           *analysis Oct 28 18:18
(base) [@splprhpc11 fastqc-analysis]$ 
(base) [@splprhpc11 fastqc-analysis]$ ll
total 3
-rwxr-x---+ 1 Domain Users 230 Oct 28 17:55 lsf-script.txt
-rwxr-x---+ 1 Domain Users 3418 Oct 28 17:55 README.md
-rwxr-x---+ 1 Domain Users 3580 Oct 28 17:55 run-fastqc-analysis.sh
(base) [splprhpc11 fastqc-analysis]$ 
(base) [splprhpc11 fastqc-analysis]$ bjobs
JOBID      USER      STAT  QUEUE   FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
272013689  ;        RUN    standard   splprhpc11 2*noderome1 *analysis Oct 28 18:18
271835463  ;        RUN    interactive svlp pondeman 2*noderome1 *OMEM_48hr Oct 27 10:56
(base) [@splprhpc11 fastqc-analysis]$ 
(base) [@splprhpc11 fastqc-analysis]$ ll
total 3
-rwxr-x---+ 1 Domain Users  0 Oct 28 18:19 job.out
-rwxr-x---+ 1 Domain Users 230 Oct 28 17:55 lsf-script.txt
-rwxr-x---+ 1 Domain Users 3418 Oct 28 17:55 README.md
-rwxr-x---+ 1 Domain Users 3580 Oct 28 17:55 run-fastqc-analysis.sh
(base) [@splprhpc11 fastqc-analysis]$ 
```

More info on scRNA data QC:

[Intro-to-scRNA](#)



# Run Modules on Interactive Mode: Launch the container

```
(base)      @splprhpc12 sc-rna-seq-snap-dnb-training]$ bsub -P hpcf_interactive -J hpcf_interactive -n 2 -q standard -R "rusage[mem=16G]" -Is "bash"
Job <270584744> is submitted to queue <standard>.
<<Waiting for dispatch ...>>
<<Starting on noderome206>>
(base)      @noderome206 sc-rna-seq-snap-dnb-training]$ singularity pull docker://achronistjude/rstudio_4.4.0_seurat_4.4.0:latest
INFO:  Using cached SIF image
(base)      @noderome206 sc-rna-seq-snap-dnb-training]$ run bash-terminal.sh
bash: run: command not found
(base)      @noderome206 sc-rna-seq-snap-dnb-training]$ bash run-terminal.sh
Apptainer> pwd
/home/
Apptainer> cd /research/
                                          /common/
```

The screenshot shows a terminal window on the left and a file explorer window on the right. The terminal window displays a command-line session for launching a Singularity container. A red arrow points from the terminal window to the file explorer window, specifically highlighting the 'run-terminal.sh' script in the file list.

The file explorer window shows the directory structure of the 'sc-rna-seq-snap-dnb-training' repository. The 'run-terminal.sh' script is highlighted with a blue selection bar at the bottom. Other files visible include 'fastqs', 'analyses', 'data', 'figures', 'LICENSE', 'package.json', 'project\_params.Config.yaml', 'README.md', 'rstudio\_4.4.0....4.0\_latest.sif', 'rstudio-container-tmp', 'run-container', 'run-rstudio.sh', and 'SECURITY.md'.

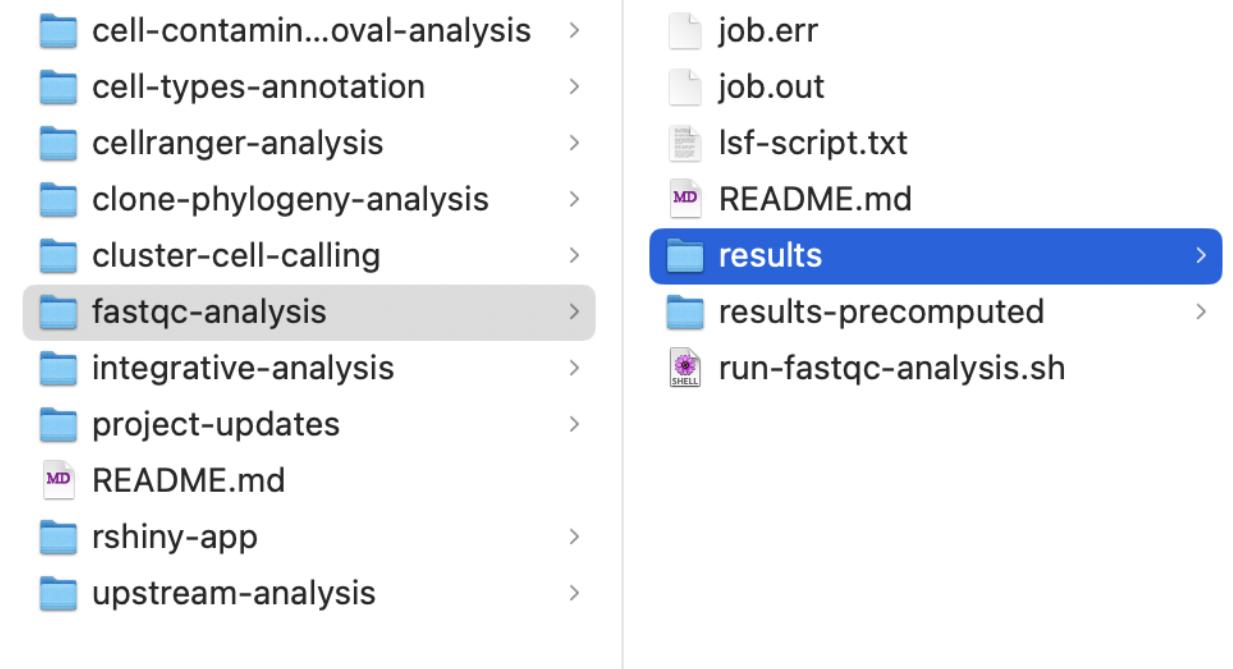
[Documentation-snap-repo-tutorial](#)



## 2. `fastqc-analysis` module

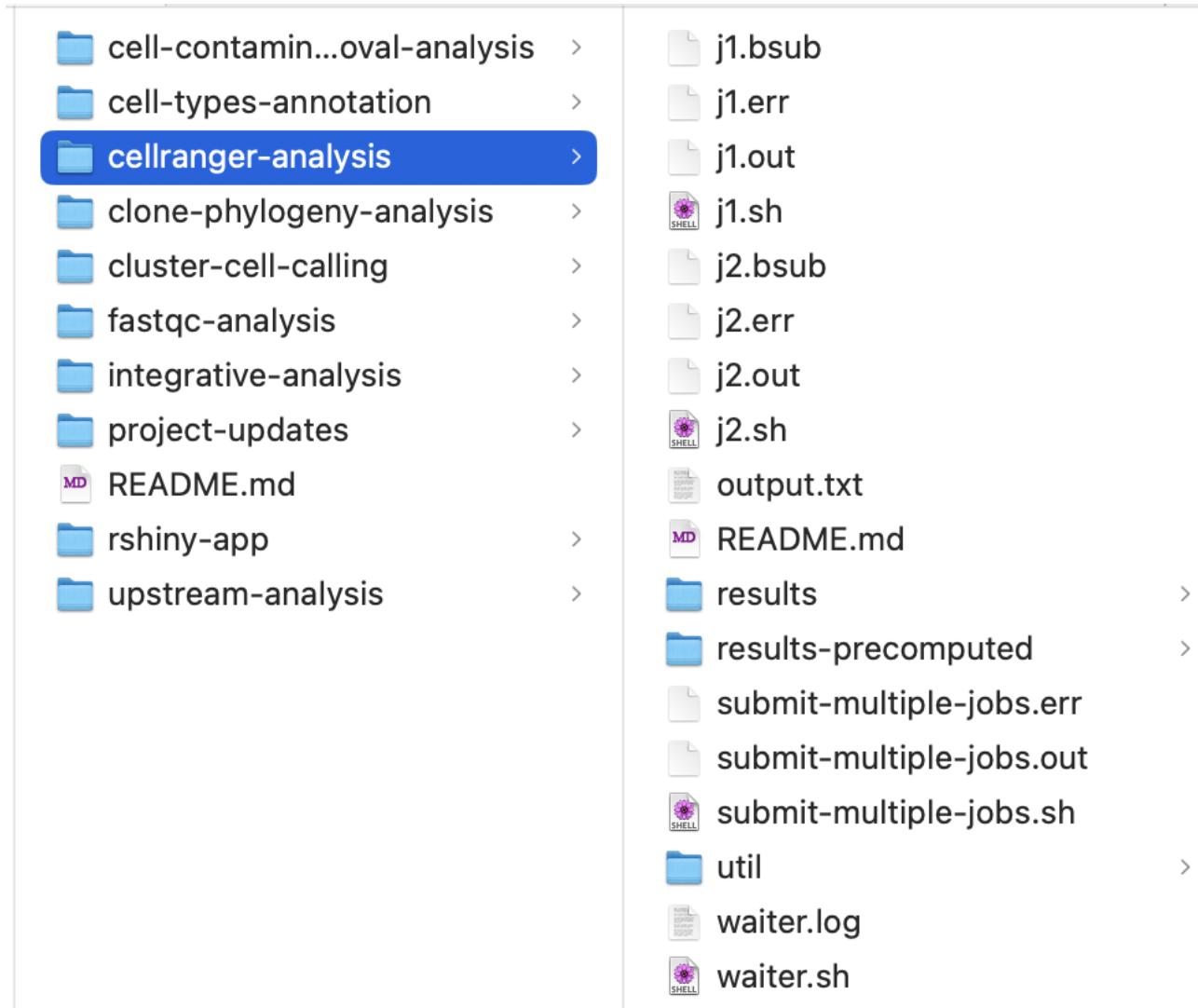
```
|Apptainer> Apptainer> cd analyses/ ←
Apptainer> ls
 README.md          cell-types-annotation  clone-phylogeny-analysis  fastqc-analysis      project-updates  upstream-analysis
cell-contamination-removal-analysis  cellranger-analysis    cluster-cell-calling     integrative-analysis  rshiny-app

Apptainer>
Apptainer>
Apptainer>
Apptainer>
Apptainer>
Apptainer> cd fastqc-analysis/
Apptainer> ls
 README.md  job.err  job.out  lsf-script.txt  results-precomputed  run-fastqc-analysis.sh
Apptainer>
Apptainer> bash run-fastqc-analysis.sh ←
/research
Metadata directory: /research/
Metadata file: project_metadata.tsv
Sample column: 1, FASTQ column: 3
Processing sample: K01, replicate: 1
  Running FastQC on: /research/
Started analysis of K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 5% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 10% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 15% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 20% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 25% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 30% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 35% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 40% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 45% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 50% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 55% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 60% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 65% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 70% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 75% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 80% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 85% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 90% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Approx 95% complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Analysis complete for K01_rep1_K01_S1_L004_R2_001.fastq.gz
Processing sample: K02, replicate: 1
```



# 3. `cellranger-analysis` module

Runtime: 40mins (training dataset) – 24hrs (8 samples; 50,000 cells)



```
(base)      @splprhpc11 analyses]$ cd cellranger-analysis/ ←
(base)      @splprhpc11 analyses]$ cd cellranger-analysis]$ ←
(base)      @splprhpc11 cellranger-analysis]$ ll
total 262
-rwxr-x---+ 1 Domain Users 651 Sep 30 11:11 j1bsub
-rwxr-x---+ 1 Domain Users 4370 Sep 28 08:58 j1.sh
-rwxr-x---+ 1 Domain Users 651 Sep 30 11:11 j2bsub
-rwxr-x---+ 1 Domain Users 1739 Sep 28 08:58 j2.sh
-rwxr-x---+ 1 Domain Users 192 Sep 30 11:12 output.txt
-rwxr-x---+ 1 Domain Users 5091 Sep 28 08:58 README.md
drwxr-x---+ 5 Domain Users 4096 Sep 29 09:52 results-precomputed
-rwxr-x---+ 1 Domain Users 1955 Sep 28 08:58 submit-multiple-jobs.sh
drwxr-x---+ 2 Domain Users 4096 Sep 28 08:58 util
-rwxr-x---+ 1 Domain Users 1628 Sep 30 11:16 waiter.log
-rwxr-x---+ 1 Domain Users 1903 Sep 28 08:58 waiter.sh
(base)      @splprhpc11 cellranger-analysis]$ ←
(base)      @splprhpc11 cellranger-analysis]$ bsub < submit-multiple-jobs.sh ←
Job <270410802> is submitted to queue <standard>.
(base)      @splprhpc11 cellranger-analysis]$ ←
(base)      @splprhpc11 cellranger-analysis]$ bjobs ←
JOBID      USER      STAT  QUEUE   FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
270315013      [REDACTED]  RUN    interactiv  svlpondeman  2*noderome1 *0MEM_48hr Sep 29 09:49
270410466      [REDACTED]  RUN    standard    noderome260  noderome272  j2          Sep 30 11:16
270410802      [REDACTED]  PEND   standard    splprhpc11          *iple-jobs Sep 30 11:17
(base)      @splprhpc11 cellranger-analysis]$
```

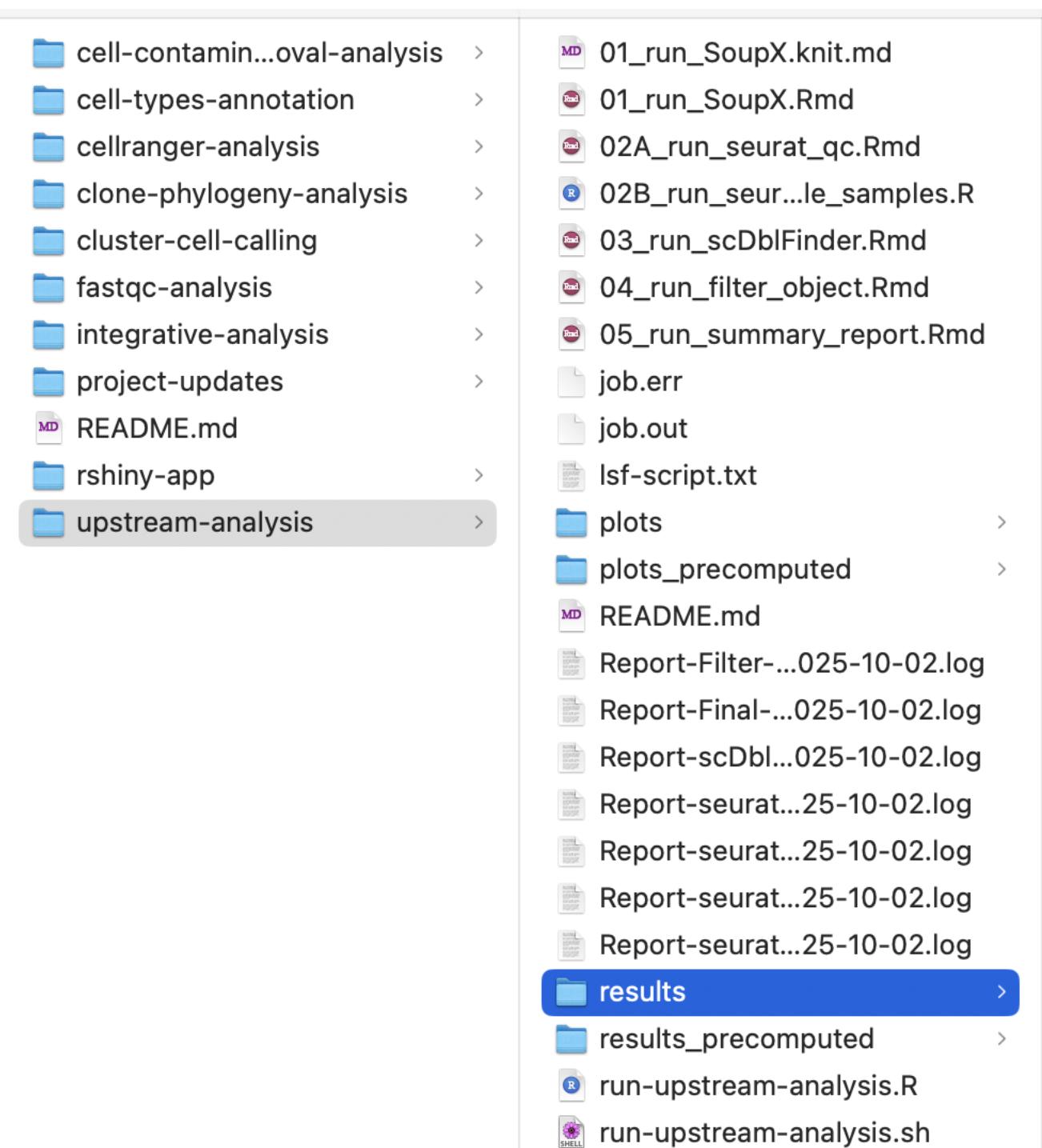
More info on scRNA data QC:

[Intro-to-scRNA](#)



# 4. `upstream-analysis` module

```
scRNA-seq-snap-pipeline-workshop]$  
scRNA-seq-snap-pipeline-workshop]$  
scRNA-seq-snap-pipeline-workshop]$ cd sc-rna-seq-snap-dnb-training/analyses/  
analyses]$  
analyses]$ cd upstream-analysis/ ←  
upstream-analysis]$  
upstream-analysis]$  
upstream-analysis]$ bsub < lsf-script.txt | ←
```



**Runtime:** 40mins (training dataset) – 6hrs (8 samples; 50,000 cells)

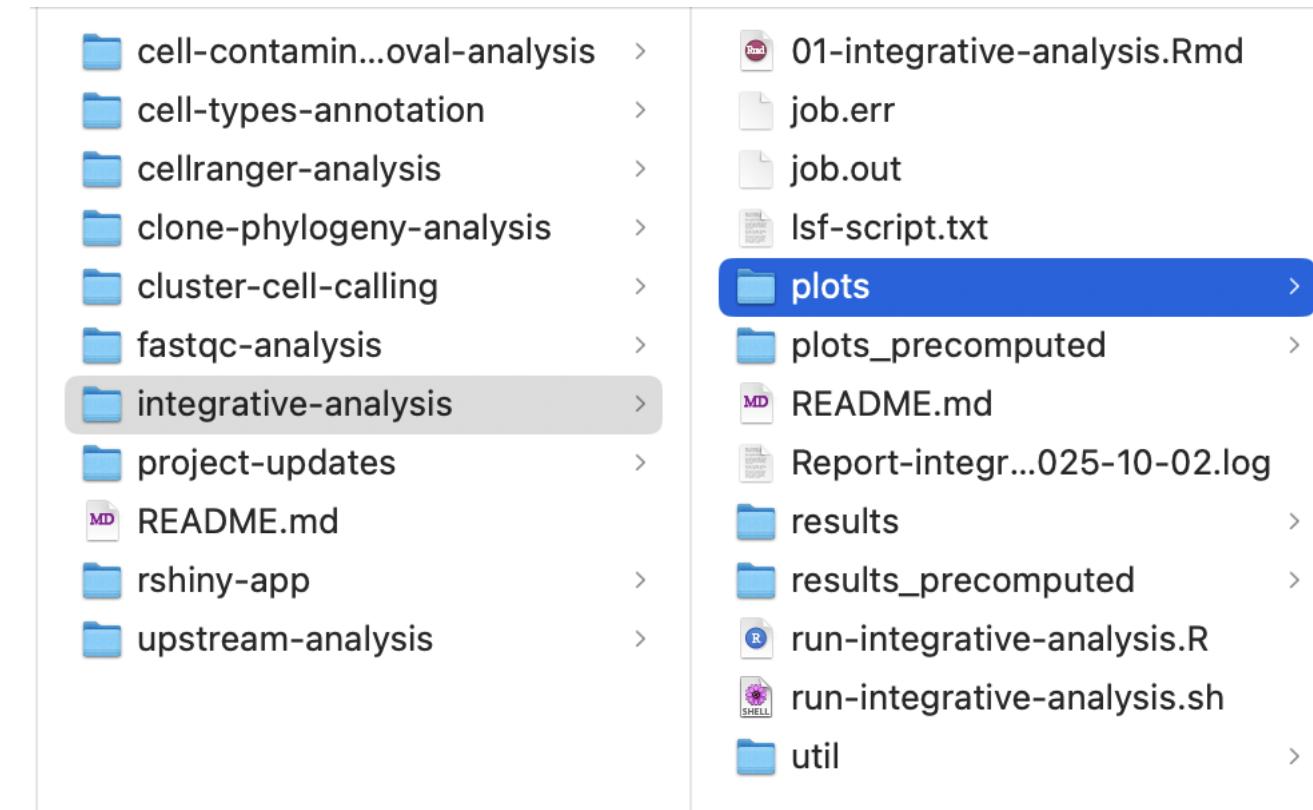
More info on scRNA data QC:

[Intro-to-scRNA](#)



# 5. `integrative-analysis` module

```
@splprhpc11 cell-types-annotation]$  
@splprhpc11 cell-types-annotation]$ cd ../integrative-analysis/ ←  
@splprhpc11 integrative-analysis]$  
@splprhpc11 integrative-analysis]$ bsub < lsf-script.txt ←
```

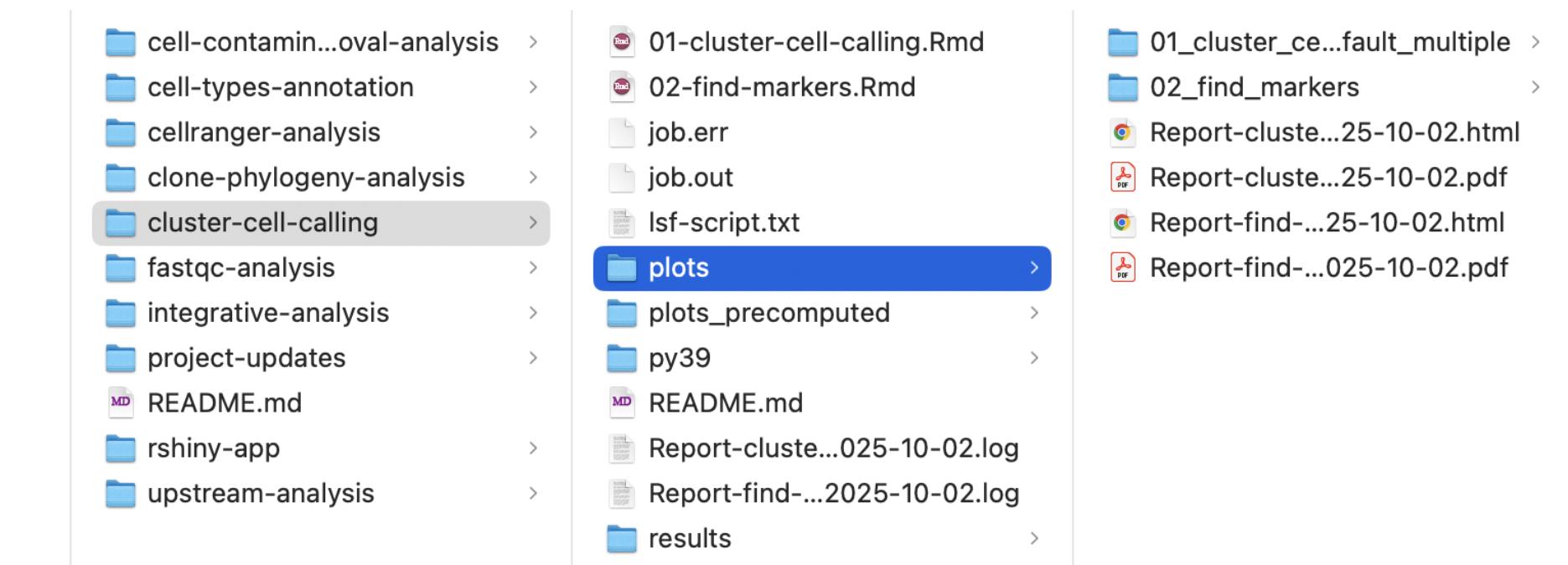


**Runtime:** 10mins (training dataset) – 2hrs (8 samples; 50,000 cells)



# 6. `cluster-cell-calling` module

```
total 20
drwxr-x---+ 2 Domain Users 4096 Sep 28 08:58 cell-contamination-removal-analysis
drwxr-x---+ 5 Domain Users 8192 Sep 30 12:31 cellranger-analysis
drwxr-x---+ 7 Domain Users 4096 Oct  2 06:51 cell-types-annotation
drwxr-x---+ 2 Domain Users 4096 Sep 28 08:58 clone-phylogeny-analysis
drwxr-x---+ 8 Domain Users 4096 Oct  1 23:58 cluster-cell-calling
drwxr-x---+ 4 Domain Users 4096 Sep 30 10:36 fastqc-analysis
drwxr-x---+ 7 Domain Users 4096 Oct  1 23:39 integrative-analysis
drwxr-x---+ 2 Domain Users 4096 Sep 28 08:58 project-updates
-rwxr-x---+ 1 Domain Users 2427 Sep 28 08:58 README.md
drwxr-x---+ 5 Domain Users 4096 Oct  2 06:58 rshiny-app
drwxr-x---+ 7 Domain Users 4096 Oct  1 23:30 upstream-analysis
(base) splprhpc12 analyses]$
(base) splprhpc12 analyses]$
(base) splprhpc12 analyses]$
(base) splprhpc12 analyses]$ cd cluster-cell-calling/ ←
(base) splprhpc12 cluster-cell-calling]$ ←
(base) splprhpc12 cluster-cell-calling]$ bsub < lsf-script.txt
```

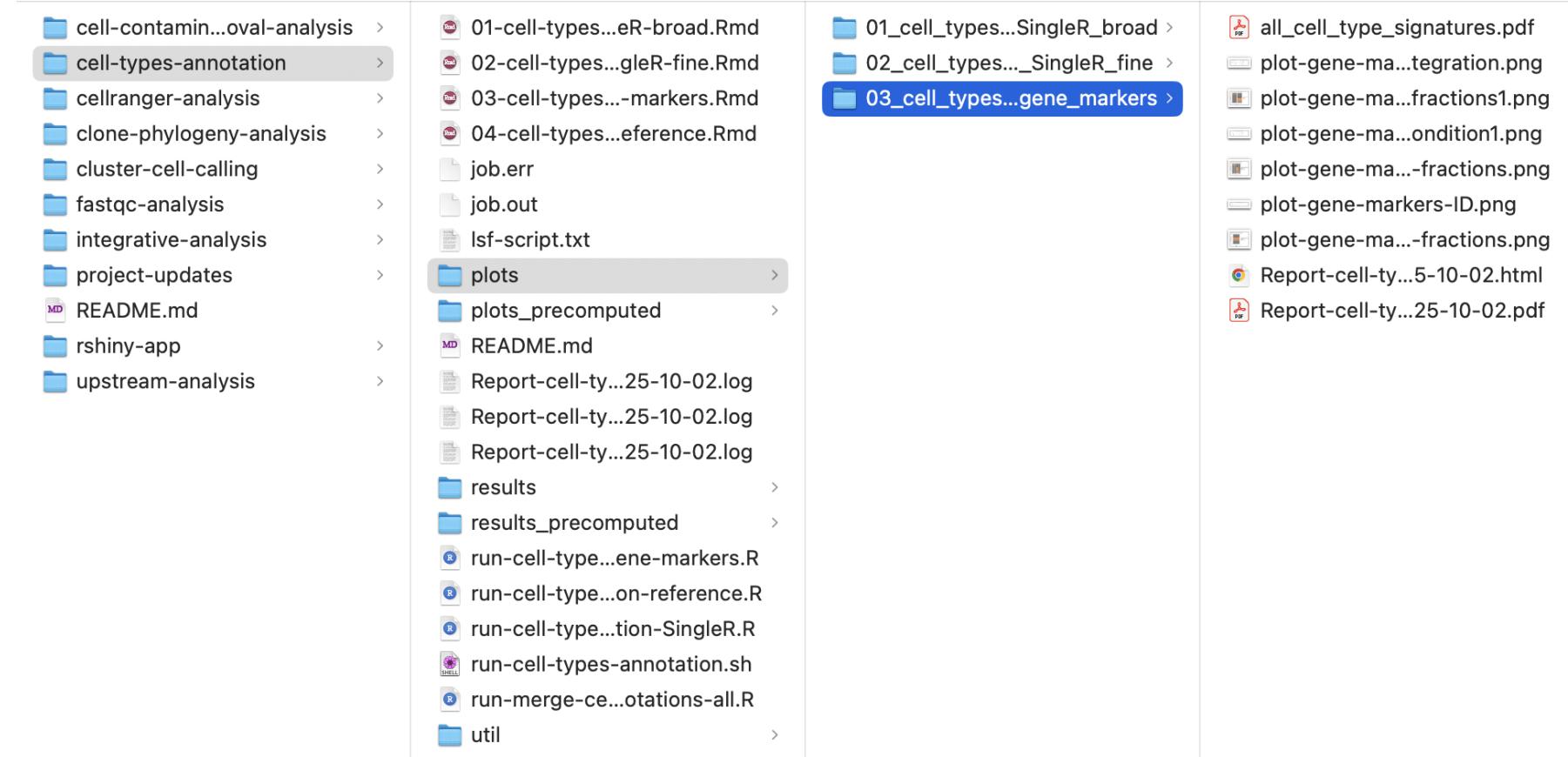


Runtime: 5mins (training dataset) – 12hrs (8 samples; 50,000 cells)



# 7. `cell-types-annotation` module

```
@splprhpc11 analyses]$ @splprhpc11 analyses]$ cd cell-types-annotation/  
@splprhpc11 cell-types-annotation]$ ls  
annotation-SingleR-broad.Rmd lsf-script.txt run-cell-types-annotation-gene-markers.R  
annotation-SingleR-fine.Rmd plots_precomputed run-cell-types-annotation-reference.R  
annotation-gene-markers.Rmd README.md run-cell-types-annotation.sh  
annotation-reference.Rmd results_precomputed run-cell-types-annotation-SingleR.R  
@splprhpc11 cell-types-annotation]$  
@splprhpc11 cell-types-annotation]$  
@splprhpc11 cell-types-annotation]$ bsub < lsf-script.txt
```

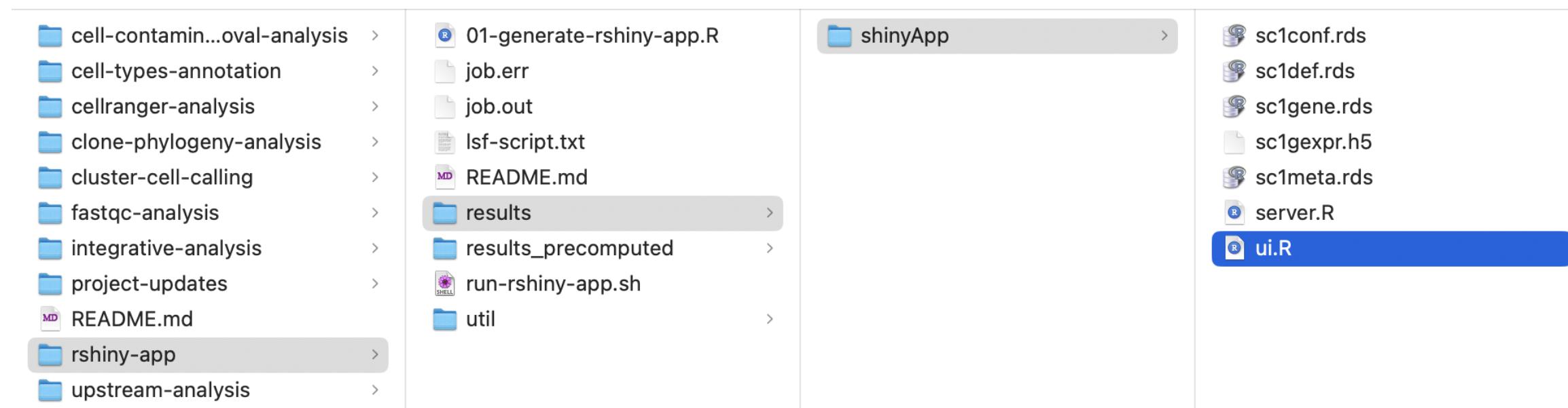


**Runtime:** 5mins (training dataset) – 3hrs (8 samples; 50,000 cells)



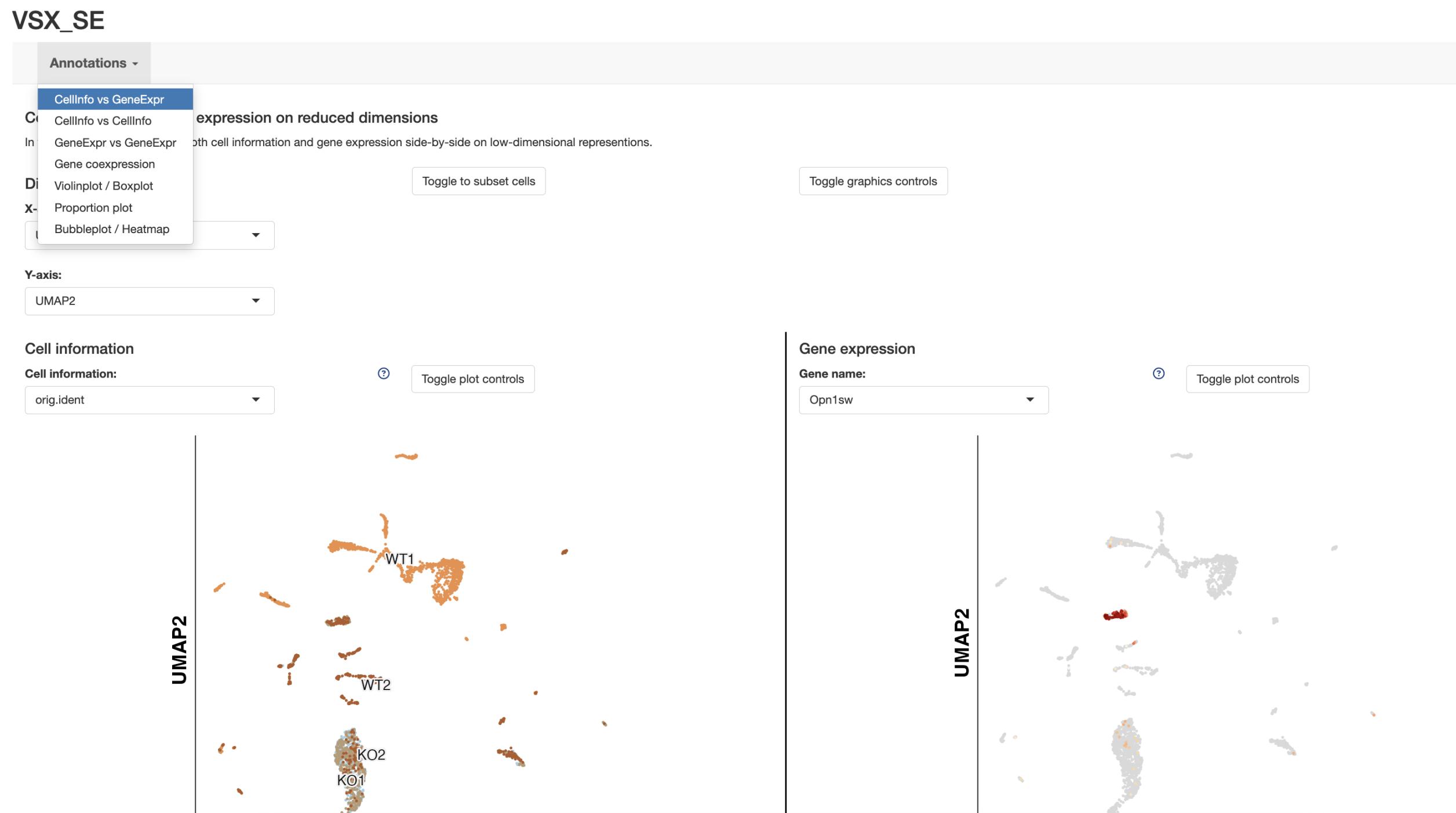
# 8. `rshiny-app` module

```
@splprhpc11 cell-types-annotation]$ cd ../rshiny-app/ ←
@splprhpc11 rshiny-app]$ ll
Domain Users 5872 Sep 28 08:58 01-generate-rshiny-app.R
Domain Users 224 Sep 28 08:58 lsf-script.txt
Domain Users 1388 Sep 28 08:58 README.md
Domain Users 4096 Sep 29 08:42 results_precomputed
Domain Users 271 Sep 28 08:58 run-rshiny-app.sh
Domain Users 4096 Sep 28 08:58 util
@splprhpc11 rshiny-app]$
@splprhpc11 rshiny-app]$
@splprhpc11 rshiny-app]$ bsub < lsf-script.txt ←
> is submitted to default queue <standard>.
@splprhpc11 rshiny-app]$
@splprhpc11 rshiny-app]$ bjobs
R STAT QUEUE FROM_HOST EXEC_HOST JOB_NAME SUBMIT_TIME
RUN interactiv svlpondeman 2*noderome1 *0MEM_48hr Oct 1 13:26
PEND standard splprhpc11 *shiny-app Oct 2 06:42
@splprhpc11 rshiny-app]$
```



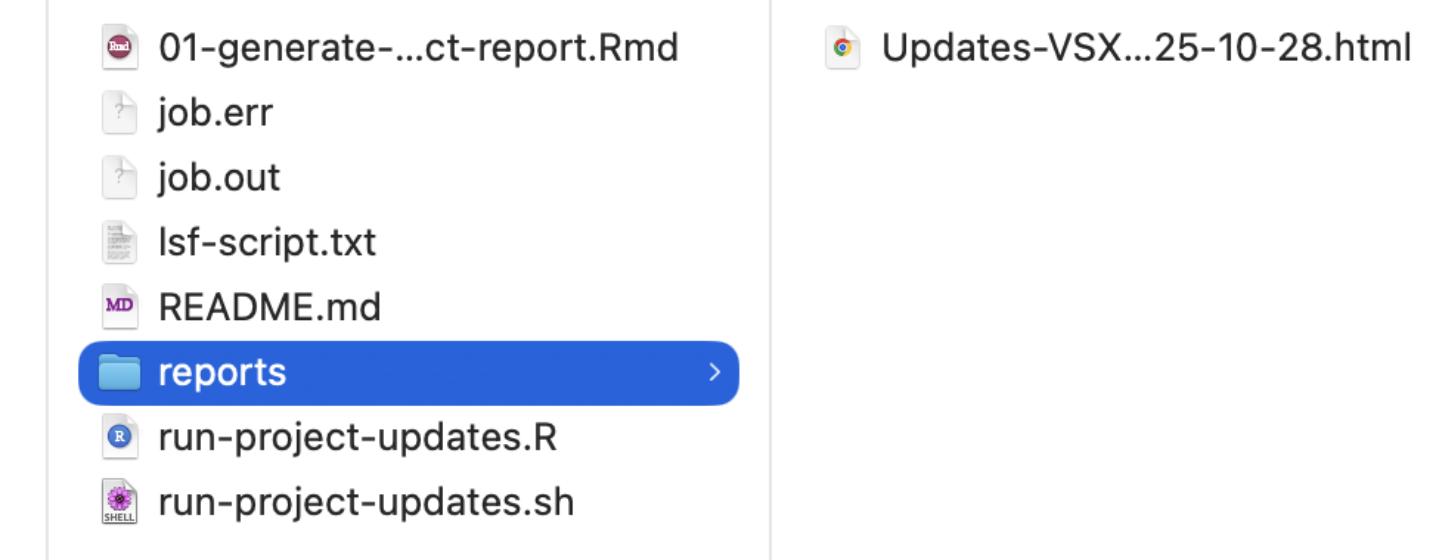
# 8. `rshiny-app` module

Runtime: 1min (training dataset) – 5mins (8 samples; 50,000 cells)



# 9. `project-updates` module

```
total 130
-rwx-----+ 1 [REDACTED] Domain Users 16464 Oct 27 19:38 01-generate-project-report.Rmd
-rwx-----+ 1 [REDACTED] Domain Users    232 Sep 16 08:57 lsf-script.txt
-rwx-----+ 1 [REDACTED] Domain Users   1939 Sep 16 08:57 README.md
-rwx-----+ 1 [REDACTED] Domain Users   3139 Sep 16 08:57 run-project-updates.R
-rwx-----+ 1 [REDACTED] Domain Users    400 Sep 16 08:57 run-project-updates.sh
(base) [REDACTED] @splprhpc08 project-updates]$
(base) [REDACTED] @splprhpc08 project-updates]$ bsub < lsf-script.txt ←
Job <271895499> is submitted to default queue <standard>.
(base) [REDACTED] @splprhpc08 project-updates]$
(base) [REDACTED] @splprhpc08 project-updates]$ bjobs
JOBID      USER      STAT  QUEUE      FROM_HOST      EXEC_HOST      JOB_NAME      SUBMIT_TIME
271835463  [REDACTED]  RUN   interactiv  svlp pondeman 2*noderome1 *0MEM_48hr Oct 27 10:56
271895499  [REDACTED]  PEND  standard    splprhpc08          *t-updates Oct 27 21:11
(base) [REDACTED] @splprhpc08 project-updates]$
(base) [REDACTED] @splprhpc08 project-updates]$ ll
total 260
-rwx-----+ 1 [REDACTED] Domain Users 16464 Oct 27 19:38 01-generate-project-report.Rmd
-rwx-----+ 1 [REDACTED] Domain Users    289 Oct 27 21:11 job.err
-rwx-----+ 1 [REDACTED] Domain Users   4064 Oct 27 21:11 job.out
-rwx-----+ 1 [REDACTED] Domain Users    232 Sep 16 08:57 lsf-script.txt
-rwx-----+ 1 [REDACTED] Domain Users   1939 Sep 16 08:57 README.md
drwx-----+ 2 [REDACTED] Domain Users  4096 Oct 27 21:11 reports
-rwx-----+ 1 [REDACTED] Domain Users   3139 Sep 16 08:57 run-project-updates.R
-rwx-----+ 1 [REDACTED] Domain Users    400 Sep 16 08:57 run-project-updates.sh
(base) [REDACTED] @splprhpc08 project-updates]$
(base) [REDACTED] @splprhpc08 project-updates]$
```



A screenshot of a bioinformatics pipeline interface. At the top, it says "Updates: VSX\_SE". Below that, it lists:

- 1 Set up
- 2 Directories and paths to file Inputs/Outputs
- 3 Methods
- 4 Results
- 5 Future directions

On the right, there's a logo for "St. Jude Children's Research Hospital" and "DNB Bioinformatics core". Below the logo, it says "Antonia Chroni for SJCRH DNB\_BINF\_Core" and "2025-10-28". A code block shows masked objects:

```
## The following objects are masked _by_ .GlobalEnv:
## 
##     PROJECT_NAME, cohort_value, genome_reference, root_dir
```

PI: Michael Dyer  
Project: VSX\_SE  
Task: NA  
Project Lead(s): NA  
Department: Developmental Neurobiology

DNB Bioinformatics Core Analysis Team:  
Lead Analyst(s): Asha Jacob Jannu  
Group Lead: Cody A. Ramirez, PhD  
Contact E-mail: [ashajacob.jannu@stjude.org](mailto:ashajacob.jannu@stjude.org)  
DNB Bioinformatics Core Pipeline: Standard sc-/sn-RNA-Seq Analysis in 10X Genomics data  
Date started: 09/16/2025

**Runtime:** 5mins (training dataset) – 10mins (8 samples; 50,000 cells)



## 10. Other available modules

---

- `cell-contamination-removal-analysis` module (description="To remove clusters and repeat steps (4) and (5), e.g. for PDX experiments.", required=False)
- `clone-phylogeny-analysis` module (description="Pipeline for Clone phylogeny analysis tool. This is currently available for human data only", required=False)



# Wrap-up: What you now have

---

- **Project folder with:** analysis/, plots/, results/
- FASTQC → Cellranger → Upstream Analysis → Integrative Analysis → Cluster Cell analysis → Cell type Annotation → R Shiny-app completed
- **Where to check success:** analyses/<module>/job.out and job.err
- **Open the app:** rshiny-app/ui.R (explore UMAPs, markers, genes)



# Takeaways from the Snap Pipeline

---

- **Dual-genome support** (e.g., human–mouse mixes) called out explicitly in the repo’s scope, handy for xenografts/chimeric contexts.
- **Breadth of built-in modules** beyond the basics, including **cell-contamination-removal** and **clone-phylogeny** analysis directories (for projects that need them).
- **Training-first packaging**: repo ships with tutorial links, one-line launchers, and a consistent folder organization -- *ideal for beginner bioinformaticians*.
- **Git-centric workflow**: fork-per-project and easy syncing with upstream; keeps analysis history, parameters, and outputs tightly versioned.



# Resources and Contact Info

---

- **Questions/Issues** related to:

**Workshop** : Asha Jacob Jannu ([ashajacob.jannu@stjude.org](mailto:ashajacob.jannu@stjude.org))

**Pipeline** : Antonia Chroni ([antonia.chroni@stjude.org](mailto:antonia.chroni@stjude.org))

Sharon Freshour ([sharon.freshour@stjude.org](mailto:sharon.freshour@stjude.org))

**All other needs or project requests**: Cody Ramirez ([cody.ramirez@stjude.org](mailto:cody.ramirez@stjude.org))

- **Tutorials** are found [here](#).
- **README**: Found in your repo root and in every module folder





# Our Tools & Resources

Tool/Resource	Description	Icon
<a href="#"><u>Snap</u></a>	sc/snRNA-Seq 10x; supports human, mouse, and dual genomes	
<a href="#"><u>Snap-10x-Flex</u></a>	sc/snRNA-Seq 10x Flex; supports human and mouse genomes	
<a href="#"><u>sc-epigenie</u></a>	scATAC-Seq 10x Flex; supports human and mouse genomes	
<a href="#"><u>sc-atac-seq-bed-builder</u></a>	Generate blacklist/promoter/enhancer BED files for analysis	
<a href="#"><u>DevOps Containers</u></a>	Reusable, independent container images for scRNA/ATAC-seq workflows	
<a href="#"><u>Trainings &amp; Workshops</u></a>	Hands-on training courses and pipeline tutorials	

